NAME (as it appears on your UF ID): ______________________________

(Please **PRINT**)

UF Student ID#: ________________

---------------------- CEN 4072/6070 Software Testing & Verification ---------------------

Exam 2 – Spring 2014

You have 90 minutes to work on this exam. It is a "closed-book/closed-notes" test. Pay attention to point values, since you may not have time to complete all 12 problems. **You should assume that all variables represent INTEGERS, unless otherwise indicated.**

PRINT your name above NOW and sign the pledge at the bottom of the last page, if appropriate, when you are finished.

PLEASE PRINT ANSWERS IN THE SPACE PROVIDED **(EXCLUDING MARGINS)** ONLY – PREFERABLY USING A BALLPOINT PEN TO INCREASE LEGIBILITY. Good luck!

1. (10 pts.) Consider a program S, pre-condition {x>0}, and post-condition {y=z+x}. Which of the following observations/facts would allow one to deduce that S is NOT ***weakly*** correct with respect to the given pre- and post-conditions. Circle either "would" or "would not" as appropriate, considering the observations individually. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

   a. For at least one initial value of x satisfying the given pre-condition, S terminates with y=z-x. — would — would not

   b. wp(S, y≠z+x) = (x<5) — would — would not

   c. sp(S, x<17) = (y≠z+x) — would — would not

   d. wlp(S, y=z) = x>5 — would — would not

   e. {x<5} S {y≠z+x} strongly — would — would not

2. a. (8 pts.) For what initial (integer) values of x and y would the program:

```
repeat
   y := y∗x
   x := x-1
until x=0
```

terminate with post-condition {y=24}? (Hint: wp(repeat S until b, Q) = $H_1 \vee H_2 \vee$... where H_1=wp(S, $b \wedge Q$), H_2=wp(S, $\neg b \wedge H_1$), etc.) Note that since x and y are assumed to be integers, limit your answer accordingly. Show your work and CIRCLE YOUR FINAL ANSWER(S).

2. (con't)

b. (10 pts.) Now consider a different program:

```
while x<>0 do
    y := y*x
    x := x-1
end_while
```

For what initial (integer) values of x and y would this program terminate with post condition {y=24}? (Hint: wp(while b do S, Q) = $H_0 \vee H_1 \vee \ldots$ where $H_0=(\neg b \wedge Q)$, $H_1=b \wedge wp(S, H_0)$, etc.) Again, limit your answer in accordance to the assumption that x and y are integers. Show your work and CIRCLE YOUR FINAL ANSWER(S).

2. (cont'd)

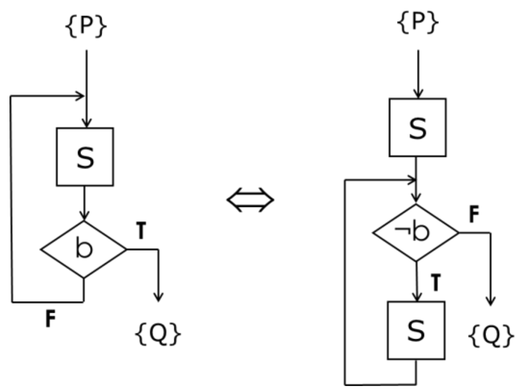
c. (4 pts.) Consider a third program:

```
y := y*x
x := x-1
while x<>0 do
    y := y*x
    x := x-1
end_while
```

For what initial (integer) values of x and y would this program terminate with post condition {y=24}? Give your answer in a form similar to that used for parts (a) and (b) above, assuming x and y are integers. Show your work and CIRCLE YOUR FINAL ANSWER(S). (Hint: note that the while_do statement in this program is identical to that used in part (b) above.)

2. (cont'd)

d. (10 pts.) Suppose for loop body S, predicate b, and loop post-condition Q, that WP_{repeat} = wp(repeat S until b, Q) and WP_{while} = wp(while **¬b** do S, Q). (NOTE the non-standard use of "**¬b**" in the while construct!) Circle "necessarily true" or "not necessarily true" for each of the following assertions as appropriate. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the item. (Hint: use whatever insights you may have gained from parts (a)-(c) above, but keep in mind that the assertions below are **general** and independent of the variable type assumption you may have employed to cull initial (x,y) values from your answers.)

i. 

necessarily true   not necessarily true

ii. $\{WP_{repeat}\}$ S $\{WP_{while}\}$ necessarily true   not necessarily true

iii. $(WP_{repeat} \wedge b) \Rightarrow Q$ necessarily true   not necessarily true

iv. $\{WP_{repeat}\}$ while ¬b do S {Q}   necessarily true   not necessarily true

v. $\{WP_{while}\}$ repeat S until b {Q}   necessarily true   not necessarily true

3. (16 pts.) Circle either "true" or "false" for each of the following assertions. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. {true} S {x>0} $\Rightarrow$ {x>0} S {x>-5} — true / false

b. {x>0} S {x<0} $\Rightarrow$ {x=17} S {x=-17} — true / false

c. {true} while x>=5 do x := x-1 {x≥4} — true / false

d. Formally speaking, Z=XJ is a **loop invariant** for the assertion: — true / false

```
{Z=0 ∧ J=0}
     while J<Y do
         Z := Z+X;
         J := J+1
     end_while
{Z=XY}
```

e. Z=XJ ∧ J≤Y is a **Q-adequate loop invariant** for the assertion given in part (d) above. — true / false

f. Suppose k = wlp(while b do S, Q). Then k is a Q-adequate loop invariant for {P} while b do S {Q} for any P that is *at least* as strong as k. — true / false

g. Using the Method of Well Founded Sets to prove that the program below will terminate for all initial values of x,y requires a simple generalization of the Method allowing measures to assume an **infinite** number of values before reaching an upper or lower bound. — true / false

```
while (y<0) do
   y := y+x
   if (x≤0) then
        x := x+1
   end_if
end_while
```

h. {P} while b do S {Q} $\Leftrightarrow$ — true / false

```
{P}
   if b then
      repeat S until NOT b
   end_if
{Q}
```

4. a. (2 pts.) Complete the following ROI for proving correctness of program S with respect to pre-condition P and post-condition Q using the weakest liberal pre-condition (wlp) predicate transform:

$$\frac{\rule{6cm}{0.4pt}}{\{P\}\ S\ \{Q\}}$$

b. (4 pts.) Give the weakest liberal pre-condition (wlp) transform rule for the *while-do* statement:

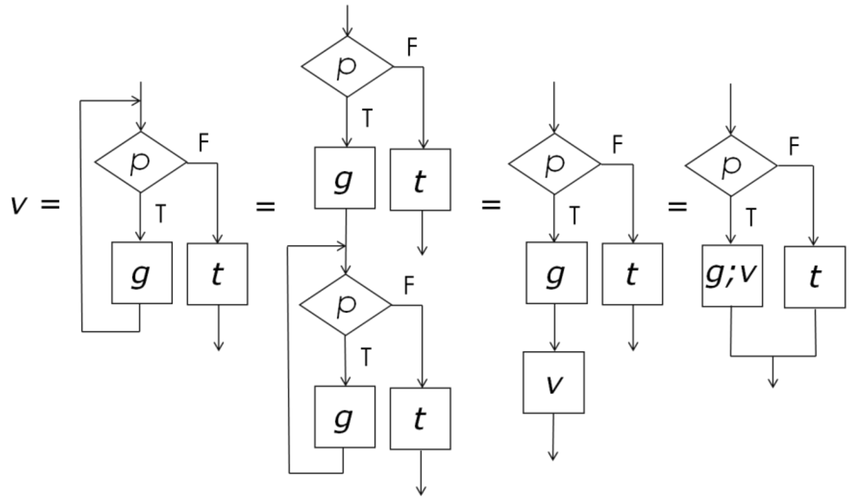
wlp(while b do S, Q) = ______________________________

c. (5 pts.) **Use the ROI from (a) and transform rule from (b) above** to prove the following assertion:

{x=3} while x<>5 do x:=x-1 {y=17}

Note: You need NOT show all proof steps. Just give the final value(s) for the term(s) you entered above for the transform rule of part (b) for the given assertion by observation (i.e., based on your understanding of what the term(s) represent) and then clearly show that the antecedent you entered for the ROI in part (a) holds.

5. (3 pts.) The diagram below was used in class to illustrate an important concept/result related to functional verification.



Which one of the following concepts/results was explained with the aid of the control flow relationships illustrated? (Circle ONE only.)

a. use of the Invariant Status Theorem (IST) to derive limited invariants

b. Subgoal Induction

c. the weakest pre-condition of *while p do g end_while; t* with respect to post-condition Q

d. the complete correctness conditions for *v* = [*while p do g; t end_while*]

e. the weakest possible *v*-adequate loop invariant for [*while p do g end_while; t*]

f. the functional relationship between *v* and a while-do construct

6. (9 pts.) Use the Axiom of Replacement and function composition to *deduce* the function of the following program:

```
if y>0 then
   x := x+1
   y := -y
end_if
y := x-1
x := 3
```

Express the function as: ($p_1 \rightarrow$ x,y := ?,? | $p_2 \rightarrow$ x,y := ?,?) where p_1 and p_2 are Boolean predicates, the union of which specifies the function domain.

program function: ____

7. (20 pts.) Circle either "valid" or "invalid" for each of the following *hypothesized* Rules of Inference. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. $\dfrac{P \Rightarrow sp(S, Q)}{\{P\}\ S\ \{Q\}}$? valid invalid

b. $\dfrac{Q \Rightarrow sp(S, P)}{\{P\}\ S\ \{Q\}}$? valid invalid

c. $\dfrac{(wlp(S, Q) \wedge K) \Rightarrow P}{\{P\}\ S\ \{Q\}}$? valid invalid

d. $\dfrac{\{P \wedge b\}\ S\ \{\neg b \wedge Q\}}{\{P\}\ \text{while } b \text{ do } S\ \{Q\}}$? valid invalid

e. $\dfrac{P \Leftrightarrow I,\ \{I\}\ S\ \{I\},\ P \Rightarrow Q}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

f. $\dfrac{b \Rightarrow Q}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

g. $\dfrac{\{true\}\ \text{if } b \text{ then } S\ \{Q\}}{\{P\}\ \text{while } b \text{ do } S\ \{Q\}}$? valid invalid

h. $\dfrac{\{P \wedge b\}\ S\ \{I\},\ ((I \vee P) \wedge \neg b) \Rightarrow Q}{\{P\}\ \text{while } b \text{ do } S\ \{Q\}}$? valid invalid

i. $\dfrac{\{true\}\ S\ \{K\},\ \{K\}\ \text{while } \neg b \text{ do } S\ \{Q\}}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

j. $\dfrac{\{true\}\ S\ \{I\},\ (I \wedge b) \Rightarrow Q}{\{P\}\ \text{repeat } S \text{ until } b\ \{Q\}}$? valid invalid

8. (18 pts.) For program *T* and intended program function *t* given below, Prove *t* = [*T*] by showing that the while_do complete correctness conditions hold. You may <u>assume</u> that the function of the loop body, *g*, is (x,y := x-1,y+2). STATE AND **PROVE** ALL OTHER CONDITIONS, STEPS, AND CASES AS ILLUSTRATED IN CLASS.

```
T:    while x<>0 do
          x := x-1;
          y := y+2
       end_while
```

$t = (x \geq 0 \rightarrow x,y := 0,y+2x)$

(Continue your proof on the next page if necessary.)

8. (cont'd)

9. Consider **intended program function** *t* from problem (8) above:

$$t = (x \geq 0 \rightarrow x,y := 0,y+2x)$$

a. (5 pts.) Derive $q(X)$, the weakest t-adequate invariant over the $D(t)$ for ***any*** while loop computing t.

b. (2 pts.) Given the initial state, $X_0 = (x_0,y_0) = (3,-2)$, what is the final state, $t(X_0)$?

c. (4 pts.) Is there *any* program of the form *while b to S* that computes *t* and produces intermediate state (5,-7) while mapping $X_0 = (3,-2)$ to $t(X_0)$? Briefly justify your answer.

d. (4 pts.) Is there *any* program of the form *while b to S* that computes *t* and produces intermediate state (8,-12) while mapping $X_0 = (3,-2)$ to $t(X_0)$? Briefly justify your answer.

e. (2 pts.) Now consider **program** ***T*** from problem (8), in which you were asked to prove $t = [T]$. *Does T* produce either of the intermediate states (5,-7) or (8,-12) while mapping $X_0 = (3,-2)$ to $t(X_0)$? Briefly explain why or why not for each.

10. (10 pts.) The following statements relate to King, et al., “Is Proof More Cost Effective than Testing?” Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. The application described in the paper is a safety critical system developed for the UK Ministry of Defense (MOD) to aid the safe operation of helicopters on naval vessels.	true	false
b. When a customer team reviewed a sample of the Z proofs (selected by them), only typographical errors were found.	true	false
c. A “lesson learned” by the authors was that at the “top” level of the system, proof annotations were often too large to be manageable, while at the “bottom” there was a need to interface with software (such as device drivers) for which there was *no* formal specification.	true	false
d. Z proofs appeared to be substantially more efficient at finding faults than the most efficient testing phase.	true	false
e. Code proofs, however, appeared to be less efficient than unit testing, because substantial amounts of unit testing were completed before the bulk of code proof started.	true	false

11. (10 pts.) The following statements relate to Linger, “Cleanroom Software Engineering for Zero-Defect Software.” Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. The Cleanroom process depends on a balance between the feature set of software and its quality. If quality is too low, then you won’t discover and learn the harder-to-find bugs because the code won't be exercised.	true	false
b. Cleanroom management planning and control is based on developing and unit-testing a pipeline of software increments that accumulate to the final product.	true	false
c. The testing employed in Cleanroom SE is biased to find errors in failure-rate order on average.	true	false
d. The philosophy of Cleanroom is that defects in software should be avoided rather than detected and repaired.	true	false
e. Linger notes that while the Cleanroom process is readily applied to new systems development, re-engineering and extension of existing systems require the use of other, more traditional processes.	true	false

12. (10 pts.) The following statements relate to functional verification and loop invariants in general, and to the Dunlop and Basili paper, "A comparative Analysis of Functional Correctness," in particular. Indicate whether each is true or false. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the problem.

a. $q(X)$ is the only f-adequate invariant over D(f) that is valid for ANY while loop that computes f and is closed for D(f).	true	false
b. In general, it is best to choose f-adequate invariants that are as weak as possible. The weaker an invariant is (while still being f-adequate), the easier it will be to use.	true	false
c. The invariant $q(X)$ occupies a unique position in the spectrum of all possible loop invariants in that it always reflects the method used by a loop on input set D(f).	true	false
d. If the function of a while loop is known, then deriving invariant $q(X)$ eliminates the need for heuristics to synthesize a Q-adequate loop invariant.	true	false
e. Verifying the properties of $q(X)$ for a given while loop, K, and hypothesized function, f, for which term(f,K) has been shown, constitutes a proof that $f = [K]$.	true	false

On my honor, I have neither given nor received unauthorized aid on this exam and I pledge not to divulge information regarding its contents to those who have not yet taken it.

____

SIGNATURE

PLEASE DO NOT BEGIN THE EXAM UNTIL TOLD TO DO SO